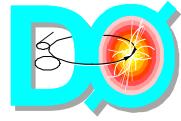


# Significant Event System Tutorial

Geoff Savage  
14 January 2002

# Outline



- 
- SES Documentation
  - Overview
  - IOC Implementation
  - SE Message Contents
  - Alarm Display
  - Configuration
    - ◆ Logger
  - Operation
    - ◆ Running
    - ◆ Files
    - ◆ Checking

# Documentation



- In the **SigEvtSys CVS repository**
  - ◆ `cvs checkout SigEvtSys`
  - ◆ `cd SigEvtSys/doc`
- DAQ shifters
  - ◆ `ses_online.pdf`
- Detector shifters and experts
  - ◆ `ses_users.pdf`
- Application developers
  - ◆ `ses_api.pdf`
- Available on the web soon!

# Overview ...



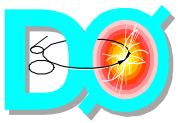
- Monitor the health of the online system
- Produce, distribute, display, and log events which are significant to the experiment
  - ◆ Alarms
  - ◆ DAQ state transitions
  - ◆ Information
- Archive events for later review
  - ◆ Help in data analysis
  - ◆ Look for trends

# ... Overview ...

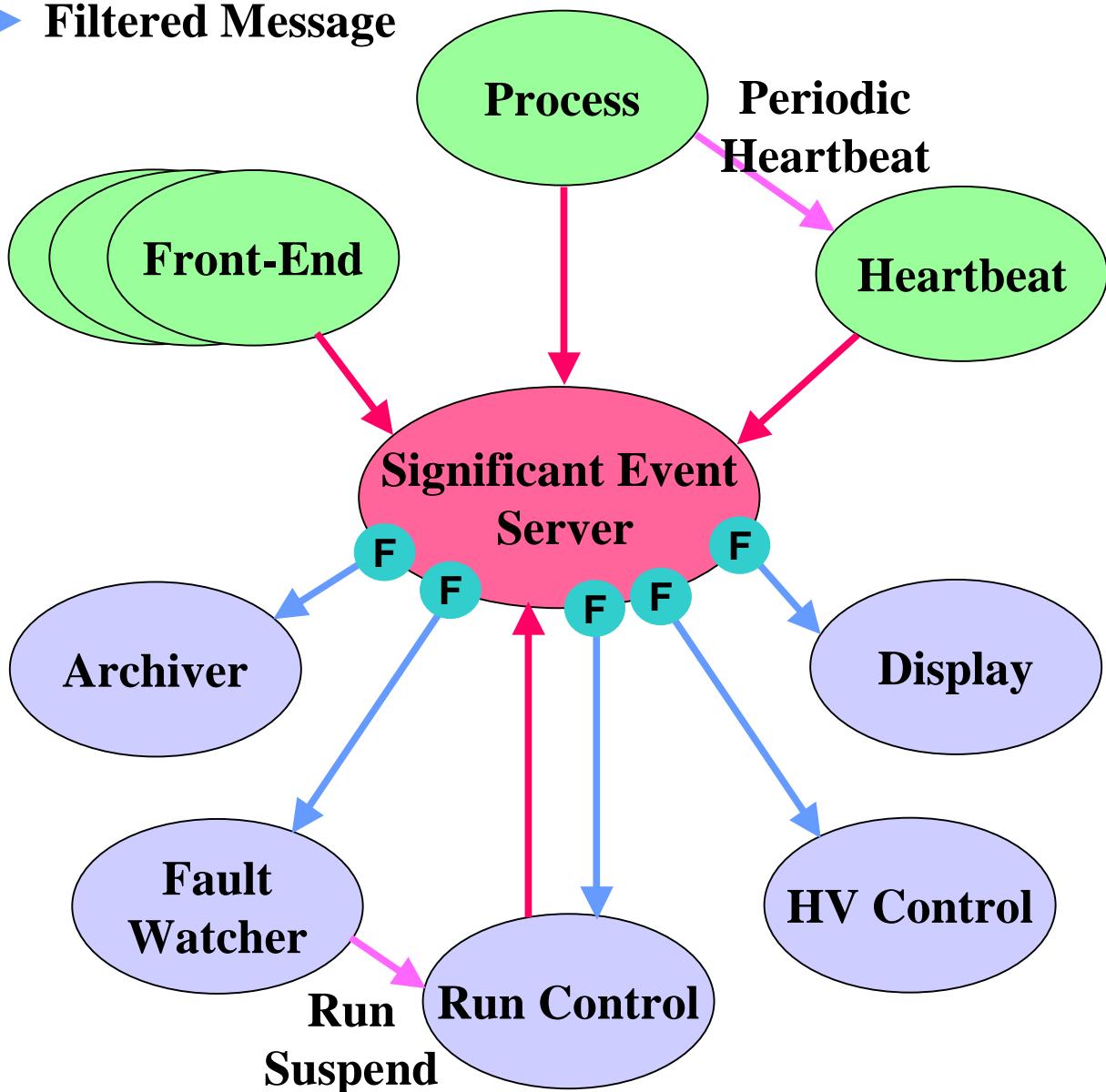


- Client/server architecture
  - ◆ Server
    - Contains a record of the current state of the online system
  - ◆ Sender clients
    - Controls system (EPICS)
    - Any (all?) online applications
  - ◆ Receiver clients
    - Filters in server restrict messages sent
    - Filter dimensions
      - name, host, priority, severity, status
    - Current state available to receiver clients when they start

# ... Overview



- F Filter
- SE Message
- Filtered Message



# IOC

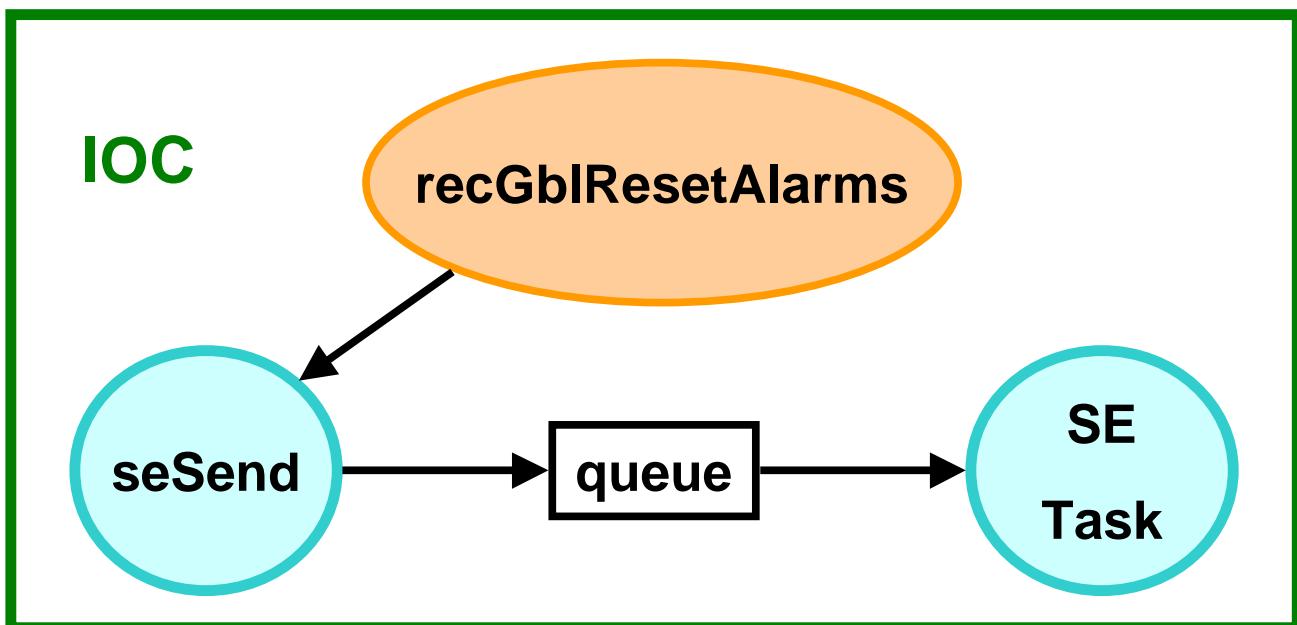


- Send messages to server when alarms are determined by EPICS
  - ◆ Set alarm limits and conditions in records
- Ai, ao, longin, longout
  - ◆ return value, limits, status (source)
- Bi, bo, mbbi, mbbo
  - ◆ return state, status
- Generic (any others)
  - ◆ return status

# seSend



- seSend called by EPICS when an alarm is determined
- Connection to seTask through a POSIX queue
- Send PV name & volatile values
  - ◆ Value, severity, status, time



# seTask



- Maintain connection to SE Server
  - ◆ Initiate connection
  - ◆ Reestablish lost connections
- Create, format, and send alarm messages to server
- Add common EPICS fields
  - ◆ Message type (info, alarm)
  - ◆ Priority - importance in DAQ
  - ◆ Alarm message type

# IOC Debugging



- Global IOC variables
- seSendState (default = 1)
  - ◆ 1 = send messages to server
  - ◆ 0 = don't send
- seTaskDebug (default = 0)
  - ◆ 1 = print messages to stdout
  - ◆ 0 = don't print messages
- Example:
  - > seSendState
  - \_seSendState = 0x3563bc: value = 1 = 0x1
  - > seSendState = 0
  - \_seSendState = 0x3563bc: value = 0 = 0x0
  - > seSendState
  - \_seSendState = 0x3563bc: value = 0 = 0x0

# IOC Config



- In startup file

extHookInit

seStart

seWait

- **extHookInit** - call **seSend**
- **seStart** - connect to the server listening on port 52150 on dOolc
- **seWait** - wait for the SES to start before starting EPICS. Otherwise alarms are detected before the server is ready.

# IOC Status



- **i - list running tasks**

NAME	ENTRY	TID	PRI	STATUS
seConnect	_seConnect	3ad5ec	149	DELAY
seTask	_seTask	3a8628	150	PEND

- **seTask** - create and send messages to the server
- **seConnect** - if connection to server is broken remind seTask to attempt to connect. Once every 30 seconds.

# Message Fields ...



- Each SE message is a string with 13 space separated fields
  - ◆ Version
  - ◆ Timestamp
  - ◆ Message type
  - ◆ Name
  - ◆ Priority
  - ◆ Host name
  - ◆ Database locator
  - ◆ Parent
  - ◆ Children
  - ◆ Transition
  - ◆ Severity
  - ◆ Alarm type
  - ◆ Parameters

# ... Message Fields ...



- Version
  - ◆ v3 python, v2 C++
  - ◆ V3 has acknowledge information not needed in C++
- Timestamp
  - ◆ Seconds since the epoch
- Message type
  - ◆ Command, alarm, info, filter, I3
- Name
  - ◆ Must follow dzero convention
- Priority
  - ◆ 0-255 - importance to experiment
- Host name
- Database locator
  - ◆ Integer value
  - ◆ Only needed if name lookups are too slow

# ... Message Fields ...



- Parent
  - ◆ For "smart" alarm acknowledgement
- Children
  - ◆ For "smart" alarm acknowledgement
- Transition
  - ◆ Bad (good -> bad)
  - ◆ Good (bad -> good)
  - ◆ Transient - notification that a problem was resolved
- Severity
  - ◆ Major, minor, invalid, no\_alarm
- Alarm type
  - ◆ Binary, comment, analog

# ... Message Fields



- Parameters

- ◆ Based on alarm type
- ◆ Binary
  - No extra data
- ◆ Comment
  - Text string
- ◆ Analog
  - Values or text
- ◆ Different message class for each alarm type

# Alarm Display ...



Group Name	MINOR	MAJOR	INVALID	ACK	GOOD
CAL	47	57	3	18	110
CFT	0	0	0	0	0
MUO	5	3	0	7	6
SMT	0	0	12	0	0
Control	8	82	0	0	36
Online	0	0	0	0	0
SDAQ	0	0	0	0	0
Alarm Watcher	0	1	0	0	0

- Each button contains the number of alarms of a severity level that pass the filter for the row
- Alarms can appear in more than one button if they pass filters for multiple rows
- Default configuration file is /online/config/ses/ad.config
  - ◆ Automatically read at startup
  - ◆ Can be substituted with -c option
  - ◆ Contains python commands that are executed

`addRow('Online', "contains(name, 'ONL')")`

`addRow('Controls', "contains(name, 'CTL')")`

# ... Alarm Display ...



- Description of columns
  - ◆ Bad alarms classified by severity
    - Minor – monitor closely
    - Major – fix a problem
    - Invalid – read or write did not complete correctly.
  - ◆ Ack – acknowledged alarms
    - Allows alarms to be ignored
    - Can unack
  - ◆ Good – after an alarm transitions from bad to good it remains in this column for five minutes then is removed

# ... Alarm Display



Group Name	MINOR	MAJOR	INVALID	ACK	GOOD
Control	22	1	0	2	0
Online	0	0	0	0	0
SDAQ	0	0	0	0	0
CFT	2	0	14	0	0
SMT	0	0	0	0	0
CAL	122	1	0	0	0
MUO	232	0	0	0	0

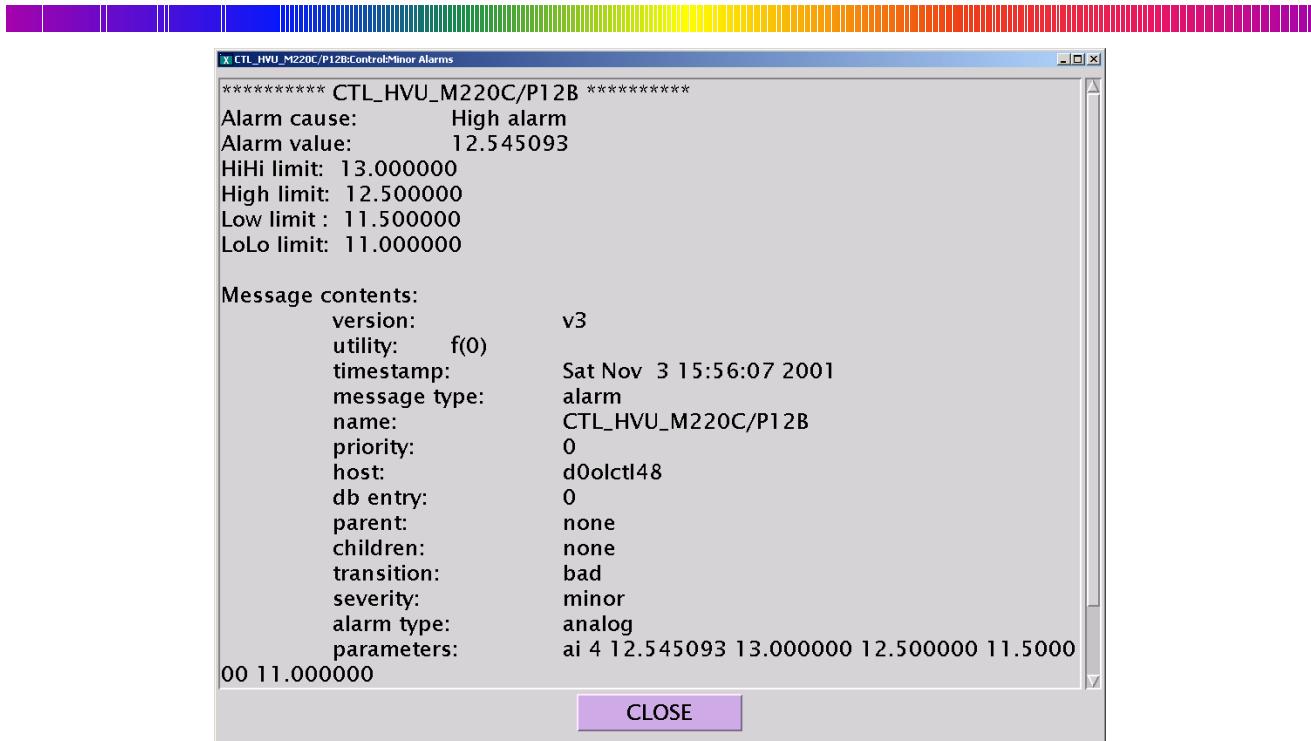
Left click a box to see the names of all the alarms in that category

CTL_HVU_M220C/P12B
CTL_HVU_M220C/M12B
CTL_HVU_M215B/M12B
CTL_HVU_M215B/M12A
CTL_HVU_M217D/M12A
CTL_HVU_M218C/P5D
CTL_HVU_M221C/M12B
CTL_HVU_M221D/P12B
CTL_HVU_M221E/P12B
CTL_HVU_M221E/M12A
CTL_HVU_M221E/M12B
CTL_HVU_M116B/M12B
CTL_HVU_M215D/M12B
CTL_HVC_07-00
CTL_HVC_07-07

SHOW  
ACK  
ACK ALL  
CLOSE

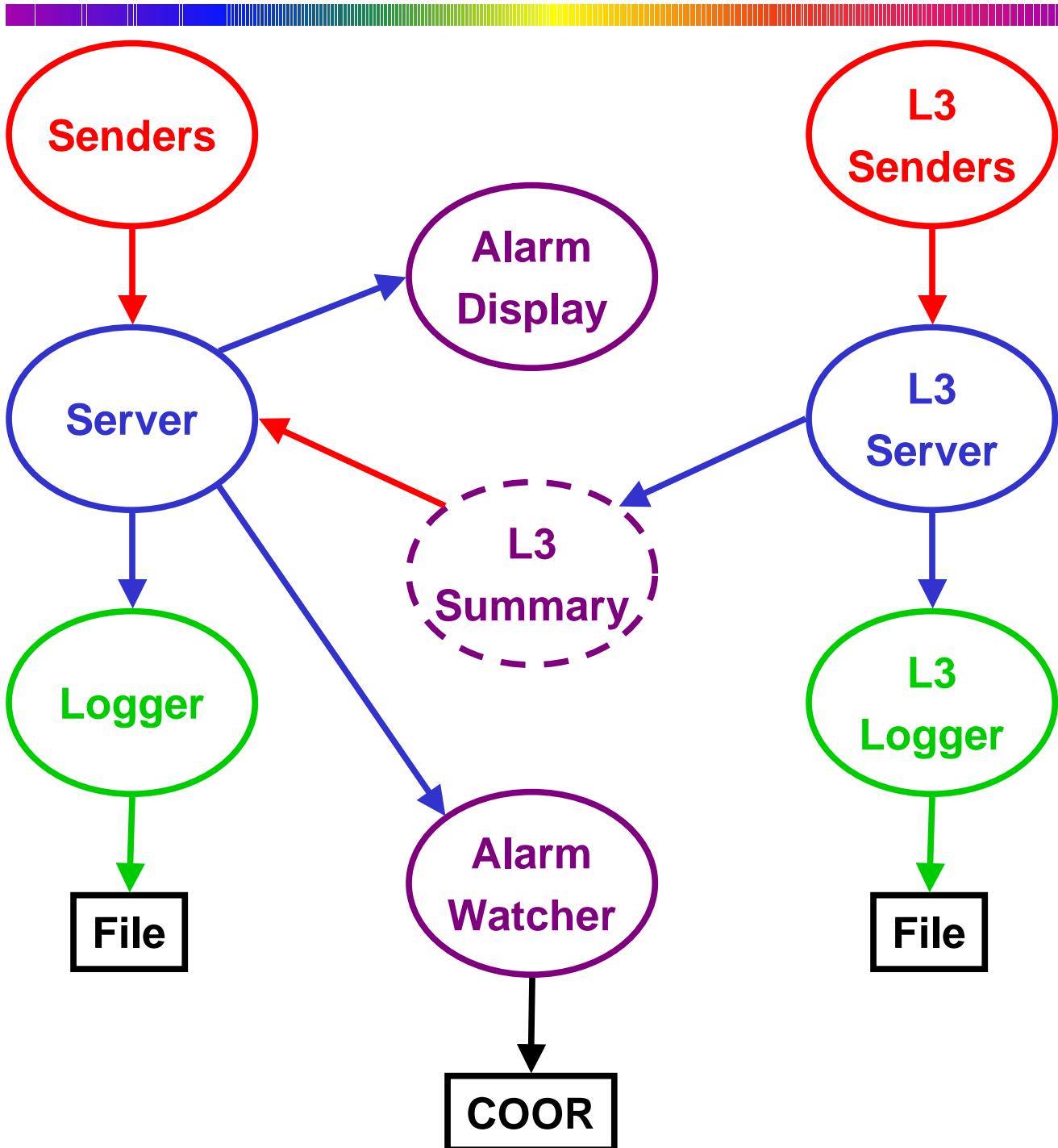
Left click a name then left click the show button to see the message display

# Message Display



- Shows details on the alarm
- The contents of all the message fields are listed
- For alarms in the good column there can be multiple alarms with the same name

# Configuration



# Logger



- Receiver client running as a daemon
- Always receives all messages
- Puts messages in log files
  - ◆ Directory /online/log/ses
  - ◆ Filenames se\_log.<date>
  - ◆ New log file at midnight
- Dies naturally when the server is stopped cleanly
- Otherwise kill by hand
  - ◆ ps aux | grep selogger
  - ◆ kill <task number>

# Running ...

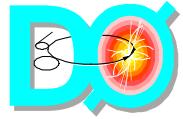


- There are two SE servers and loggers that run as daemons
  - ◆ Standard
  - ◆ Level 3
    - Handles enormous number of L3 error messages
- All are started with
  - > start\_daq ses
- All are stopped
  - > stop\_daq ses
- Start displays on console
  - > start\_daq screen3

# ... Running

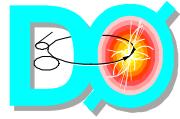


- The server and apps need to know
  - ◆ On which host the server is executing
  - ◆ On which port the server is listening for connection requests
- Environment variables in **SigEvtSys** package
  - `SES_SERVER_PORT`
  - `SES_SERVER_HOST`
  - `SES_LOGGER_HOST`
  - `SES_L3_SERVER_PORT`
  - `SES_L3_SERVER_HOST`
  - `SES_L3_LOGGER_HOST`



# Files

- Official log files in  
`/online/log/ses`
- L3 log files are in  
`/online/log/ses/l3`
- Both sets of server error files are in  
`/online/log/ses/server`
- Both sets of logger error files in  
`/online/log/ses/logger`



# Checking

- Names of the daemons
  - ◆ **seserver**
    - SES server on port 52150
    - L3 SES server on port 52154
  - ◆ **selogger**
- Log into host where they are executing and see if the daemon is executing
  - > `ps auxwww | grep seserver`
  - > `ps auxwww | grep selogger`